

Prediction of power in solar stirling heat engine by using neural network based on hybrid genetic algorithm and particle swarm optimization

Mohammad Hossien Ahmadi · Saman Sorouri Ghare Aghaj ·
Alireza Nazeri

Received: 13 November 2011 / Accepted: 1 February 2012 / Published online: 25 February 2012
© Springer-Verlag London Limited 2012

Abstract In this paper, the model based on a feed-forward artificial neural network optimized by particle swarm optimization (HGAPSO) to estimate the power of the solar stirling heat engine is proposed. Particle swarm optimization is used to decide the initial weights of the neural network. The HGAPSO-ANN model is applied to predict the power of the solar stirling heat engine which data set reported in literature of china. The performance of the HGAPSO-ANN model is compared with experimental output data. The results demonstrate the effectiveness of the HGAPSO-ANN model.

Keywords Solar dish · Stirling heat engine · Artificial neural network · Particle swarm optimization · Genetic algorithm · Hybrid

List of symbols

C_V	Specific heat capacity, $\text{J mol}^{-1} \text{K}^{-1}$
h	Heat transfer coefficient, WK^{-1} or WK^{-4} or $\text{Wm}^{-2} \text{K}^{-1}$
n	The mole number of the working fluid, mol
M	Regenerative time constant, Ks^{-1}
p	Power, W
Q	Heat transfer, J
R	The gas constant, $\text{J mol}^{-1} \text{K}^{-1}$
t	Time, s
T	Temperature, K

W	Work, J
λ	Ratio of volume during the regenerative processes
τ	Cyclic period, s
η	Thermal efficiency
k_0	Heat leak coefficient, WK^{-1}
ε_R	Effectiveness of the regenerator

Subscripts

H	Absorber
HC	High temperature side convection
HR	High temperature side radiation
L	Heat sink
LC	Low temperature side convection
m	The system
R	Regenerator
t	Stirling engine
0	Ambient or optics
4-1	The processes

1 Introduction

Solar thermal power systems utilize the heat generated by a collector concentrating and absorbing the sun's energy to drive a heat engine/generator and produce electric power. Of the three solar thermal systems, the tower, the trough, and the dish, the dish-stirling systems have demonstrated the highest efficiency [1–3].

Over the last 20 years, eight different dish-stirling systems ranging in size from 2 to 50 kW have been built by companies in the United states, Germany, Japan, and Russia [1]. In principle, high concentrating and low or nonconcentrating solar collectors can all be used to power the stirling engine.

Dish-stirling solar thermal power bases on spot-focused collecting heat and stirling thermodynamics cycle.

M. H. Ahmadi (✉)
Faculty of Mechanical Engineering, K.N. Toosi University,
Pardis Ave, Mollasadra Street, Vanak Sq, Tehran, Iran
e-mail: mohammadhosein.ahmadi@gmail.com

S. Sorouri Ghare Aghaj · A. Nazeri
Department of Industrial Engineering, Science and Research
Branch, Islamic Azad University, Hormozgan, Iran

The concentration ratio of dish concentrator can reach 1,500–3,000, and the operating temperature of the receiver can reach more than 800°C. Reported from literatures, the peak solar-to-electric conversion efficiency of the system has reached 29.4%. Because dish-stirling solar thermal power system adopts the idea of modularity generally, it can be used as scattered power source of electric power system; also it can make up dish field to provide electric power assembly. Just as its high efficiency and flexible combinations, it has been developed quickly in developed countries in the last 20 years, and the output power of unit system ranged from 2 to 50 kWe.

A soft sensor is a conceptual device whose output or inferred variable can be modeled in terms of other parameters that are relevant to the same process [4]. According to Rallo et al. [4], artificial neural network could be used as soft sensor building approach. The ANN is a popular, nonlinear, and nonparametric tool in oil flow rate estimation.

In the present work, HGAPSO is proposed for optimizing the weights of feed-forward neural network. Then, simulation results demonstrate the effectiveness and potential of the new proposed network for power prediction of solar dish-stirling engine that reported in literature in china [5] compared with experimental data.

2 Stirling engine

In order to produce electricity using concentrated solar energy, a proper power cycle that converts the thermal energy to mechanical work is necessary. As well known, stirling engine uses externally supplied thermal energy to make a mechanical motion using repeated heating/cooling of working gas like helium or hydrogen in a closed loop. So the engine could run by any sources of thermal energy as long as the energy is properly transferred into the working gas. Owing to this advantage, Stirling engine has been taken much interest from various renewable energy technologies. Additionally, the engine is known to make relatively low noise compared to the internal combustion engines and also more environmentally friendly in terms of air pollution. Solar application of stirling engine, especially for a small-scale dish-type power system, is one of promising applications, which is already in a commercial stage through over 20 years of R and D efforts [6].

The dish-stirling system, comprising a parabolic dish collector (which is made up a dish concentrator and a thermal absorber) and a stirling heat engine located at the focus of the dish, tracks the sun and focuses solar energy into a cavity absorber where solar energy is absorbed and transferred to the stirling engine to heat its displacer hot-end, thereby creating a solar-powered stirling heat engine, as shown in Fig. 1.

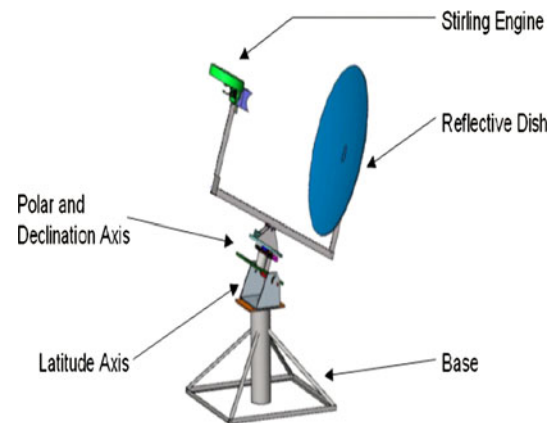


Fig. 1 Schematic diagram of the dish system

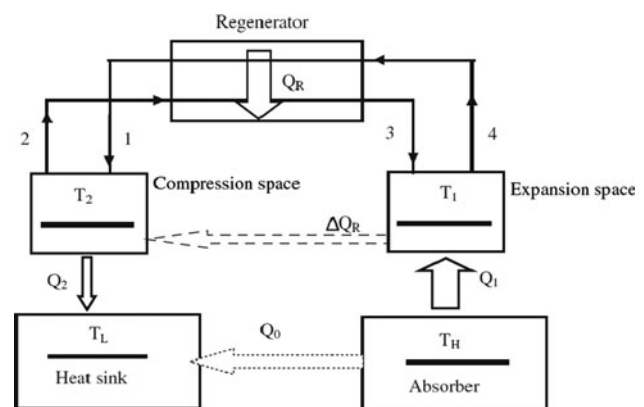


Fig. 2 Schematic diagram of the stirling heat engine cycle

Figure 2 is a schematic diagram of a stirling heat engine cycle with finite-time heat transfer and regenerative heat losses as well as conductive thermal bridging losses from the absorber to the heat sink. This cycle approximates the compression stroke of real heat engine as an isothermal process 1–2, with an irreversible heat rejection at constant temperature T_2 to the heat sink at constant temperature T_L . The heat addition to the working fluid from the regenerator is modeled as isochoric process 2–3. The expansion stroke producing work is modeled as isothermal process 3–4, with irreversible heat addition at constant temperature T_1 from the absorber at constant temperature T_H . Finally, process 4–1 closed the cycle, the heat is rejected to the regenerator is modeled as isochoric process 4–1.

If the regenerator is ideal, the heat absorbed during process 4–1 should be equal to the heat rejected during process 2–3; however, the ideal regenerator requires an infinite area or infinite regeneration time to transfer finite heat amount, and this is impractical. Therefore, it is desirable to consider a real regenerator with heat losses ΔQ_R . In addition, we also consider conductive thermal bridging losses Q_0 from the absorber to the heat sink.

3 Methodology

3.1 Finite-time thermodynamic analysis for solar-powered stirling heat engine

Chambadal and novikov first applied finite-time thermodynamics to analyze heat engine in 1975, since then, extensive research have been undertaken on the performance analysis and optimization of stirling heat engines and solar-powered stirling engines based on FTT [7].

A part of the above publications is dedicated to the performance analysis and optimization of low temperature differential stirling heat engines powered by low concentrating solar collectors [8].

3.1.1 Regenerative heat losses of the regenerator

It is important to mention that there also exists a finite heat transfer in the regenerative heat transfer (Q_R) that is given by [9, 10]:

$$Q_R = nC_V \varepsilon_R (T_1 - T_2) \quad (1)$$

where C_V is the specific heat of working substance and ε_R is the effectiveness of the regenerator. Thus, the regenerative heat loss in the two regenerative processes is given by [9, 10]:

$$\Delta Q_R = nC_V (1 - \varepsilon_R) (T_1 - T_2) \quad (2)$$

In order to calculate the time of the regenerative processes, one assumes that the temperature of the working fluid in the regenerative processes as a function of time is given by [10]:

$$\frac{dT}{dt} = \pm M_i \quad (3)$$

where M is the proportionality constant that is independent of the temperature but dependent on the property of the regenerative material. The positive and negative signs correspond to constant volume heating ($i = 1$) and cooling ($i = 2$) processes, respectively.

One obtains the time of the two isochoric processes as:

$$t_3 = \frac{T_1 - T_2}{M_1} \quad (4)$$

$$t_4 = \frac{T_1 - T_2}{M_2} \quad (5)$$

3.1.2 The conductive thermal bridging losses from the absorber to the heat sink

The conductive thermal bridging losses from the absorber at temperature T_H to the heat sink at temperature T_L is assumed to be proportional to the cycle time and given by [10, 11]:

$$Q_0 = k_0 (T_H - T_L) \tau \quad (6)$$

where k_0 is the heat leak coefficient between the absorber and the heat sink, and τ is the cyclic period.

3.1.3 The amounts of heat released by absorber and absorbed by the heat sink

For a stirling cycle, the amounts of heat released by the absorber and absorbed by heat sink are as follows [9]:

$$Q_1 = [h_{HC}(T_H - T_1) + h_{HR}(T_H^4 - T_1^4)]t_1 = nRT_1 Ln\lambda + nC_V(1 - \varepsilon_R)(T_1 - T_2) \quad (7)$$

$$Q_2 = h_{LC}(T_2 - T_L)t_2 = nRT_2 Ln\lambda + nC_V(1 - \varepsilon_R)(T_1 - T_2) \quad (8)$$

Let Q_1 be the amount of heat absorbed by the working fluid at temperature T_1 from the absorber at temperature T_H .

Let Q_2 be the amount of heat released by working fluid at temperature T_2 to the heat sink at temperature T_L .

Where h_{HC} is high temperature side convection heat transfer coefficient, h_{HR} is high temperature side radiation heat transfer coefficient, h_{LC} is low temperature side convection heat transfer coefficient, R is the gas constant, and λ is the ratio of volume during the regenerative processes, that is,

$$\lambda = \frac{V_1}{V_2} \quad (9)$$

Take into account the major irreversibility mentioned above, the net heats released from the absorber Q_H and absorbed by the heat sink Q_L are given as:

$$Q_H = Q_1 + Q_0 \quad (10)$$

$$Q_L = Q_2 + Q_0 \quad (11)$$

3.1.4 The cyclic period

Using Eqs. 4–8, we get that the cyclic period τ is:

$$\begin{aligned} \tau &= t_1 + t_2 + t_3 + t_4 \\ &= \frac{nRT_1 Ln\lambda + nC_V(1 - \varepsilon_R)(T_1 - T_2)}{[h_{HC}(T_H - T_1) + h_{HR}(T_H^4 - T_1^4)]} \\ &\quad + \frac{nRT_2 Ln\lambda + nC_V(1 - \varepsilon_R)(T_1 - T_2)}{h_{LC}(T_2 - T_L)} \\ &\quad + \left(\frac{1}{M_1} + \frac{1}{M_2} \right) \end{aligned} \quad (12)$$

3.1.5 Maximum power output and maximum power efficiency of the stirling engine

The power output and the thermal efficiency are given by [7]:

$$p = \frac{W}{\tau} = \frac{Q_H - Q_L}{\tau} \quad (13)$$

$$\eta_i = \frac{Q_H - Q_L}{Q_H} \quad (14)$$

Using Eqs. 6–14, we have:

$$P = \frac{T_1 - xT_1}{\frac{T_1 + A_1(T_1 - xT_1)}{h_{HC}(T_H - T_1) + h_{HR}(T_H^4 - T_1^4)} + \frac{xT_1 + A_1(T_1 - xT_1)}{h_{LC}(xT_1 - T_L)} + F_1(T_1 - xT_1)} \quad (15)$$

$$\eta_i = \frac{T_1 - xT_1}{T_1 + A_1(T_1 - xT_1) + [k_0(T_H - T_L)] \left[\frac{T_1 + A_1(T_1 - xT_1)}{h_{HC}(T_H - T_1) + h_{HR}(T_H^4 - T_1^4)} + \frac{xT_1 + A_1(T_1 - xT_1)}{h_{LC}(xT_1 - T_L)} + F_1(T_1 - xT_1) \right]} \quad (16)$$

where

$$A_1 = \frac{C_V(1 - \varepsilon_R)}{RLn\lambda}$$

and

$$F_1 = \frac{1}{nRLn\lambda} \left(\frac{1}{M_1} + \frac{1}{M_2} \right)$$

and

$$x = T_2/T_1$$

Therefore, the maximum power output and the corresponding optimal thermal efficiency of the stirling engine are:

$$P_{\max} = \frac{1 - x}{\frac{1 + A_1(1 - x)}{h_{HC}(T_H - T_{1opt}) + h_{HR}(T_H^4 - T_{1opt}^4)} + \frac{x + A_1(1 - x)}{h_{LC}(xT_{1opt} - T_L)} + F_1(1 - x)} \quad (17)$$

$$\eta_{1opt} = \frac{1 - x}{1 + A_1(1 - x) + [k_0(T_H - T_L)] \left[\frac{1 + A_1(1 - x)}{h_{HC}(T_H - T_{1opt}) + h_{HR}(T_H^4 - T_{1opt}^4)} + \frac{x + A_1(1 - x)}{h_{LC}(xT_{1opt} - T_L)} + F_1(1 - x) \right]} \quad (18)$$

3.2 Artificial neural networks

Artificial neural networks are parallel information processing methods that can express complex and nonlinear relationship use number of input–output training patterns from the experimental data. ANNs provides a nonlinear mapping between inputs and outputs by its intrinsic ability [12]. The success in obtaining a reliable and robust network depends on the correct data preprocessing, correct architecture selection, and correct network training choice strongly [13].

The most common neural network architecture is the feed-forward neural network. Feed-forward network is the network structure in which the information or signals will propagates only in one direction, from input to output. A three-layered feed-forward neural network with back propagation algorithm can approximate any nonlinear continuous function to an arbitrary accuracy [12, 14].

The network is trained by performing optimization of weights for each node interconnection and bias terms, until the values output at the output layer neurons are as close as possible to the actual outputs. The mean squared error of the network (MSE) is defined as:

$$MSE = \frac{1}{2} \sum_{k=1}^G \sum_{j=1}^m [Y_j(k) - T_j(k)]^2 \quad (19)$$

where m is the number of output nodes, G is the number of training samples, $Y_j(k)$ is the expected output, and $T_j(k)$ is the actual output.

The data are split into two sets, a training data set and a validating data set. The model is produced using only the training data. The validating data are used to estimate the accuracy of the model performance. In training a network, the objective is to find an optimum set of weights. When the number of weights is higher than the number of available data, the error in-fitting the nontrained data initially decreases

but then increases as the network becomes over-trained. In contrast, when the number of weights is smaller than the number of data, the over-fitting problem is not crucial.

4 Genetic algorithms and particle swarm optimization

4.1 Particle swarm optimization

Particle swarm optimization (PSO) is one of the recent evolutionary optimization methods. This technique was

originally developed by Kennedy and Eberhart [15] in order to solve problems with continuous search space. PSO is based on the metaphor of social interaction and communication, such as bird flocking and fish schooling. This algorithm can be easily implemented and it is computationally inexpensive, since its memory and CPU speed requirements are low [16].

PSO shares many common points with GA. It conducts the search using a population of particles that correspond to individuals in GA. Both algorithms start with a randomly generated population. PSO does not have a direct recombination operator. However, the stochastic acceleration of a particle toward its previous best position, as well as toward the best particle of the swarm (or toward the best in its neighborhood in the local version), resembles the recombination procedure in evolutionary computation [17–19].

Compared to GA, the PSO has some attractive characteristics. It has memory, so knowledge of good solutions is retained by all particles, whereas in GA, previous knowledge of the problem is destroyed once the population changes. PSO does not use the filtering operation (such as selection in GAs), and all the members of the population are maintained through the search procedure to share their information effectively.

PSO uses social rules to search in the design space by controlling the trajectories of a set of independent particles. The position of each particle, x_i , representing a particular solution of the problem, is used to compute the value of the fitness function to be optimized. Each particle may change its position and consequently may explore the solution space, simply varying its associated velocity. In fact, the main PSO operator is the velocity update, which takes into account the best position, in terms of fitness value reached by all the particles during their paths, P_g^t , and the best position that the agent itself has reached during its search, P_i^t , resulting in a migration of the entire swarm toward the global optimum.

At each iteration, the particle moves around according to its velocity and position; the cost function to be optimized is evaluated for each particle in order to rank the current location. The velocity of the particle is then stochastically updated according to

$$V_i^{t+1} = \omega V_i^t + C_1 r_1^t (P_i^t - X_i^t) + C_2 r_2^t (P_g^t - X_i^t) \quad (20)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (21)$$

Equation 2 describes how the velocity is dynamically updated and Eq. 3 is used to update the position of the “flying” particles.

V_i^t is the velocity vector at iteration t , r_1 and r_2 represent random numbers in the range $[0,1]$; P_g^t denotes the best ever particle position of particle i , and P_i^t corresponds to the global best position in the swarm up to iteration t [17].

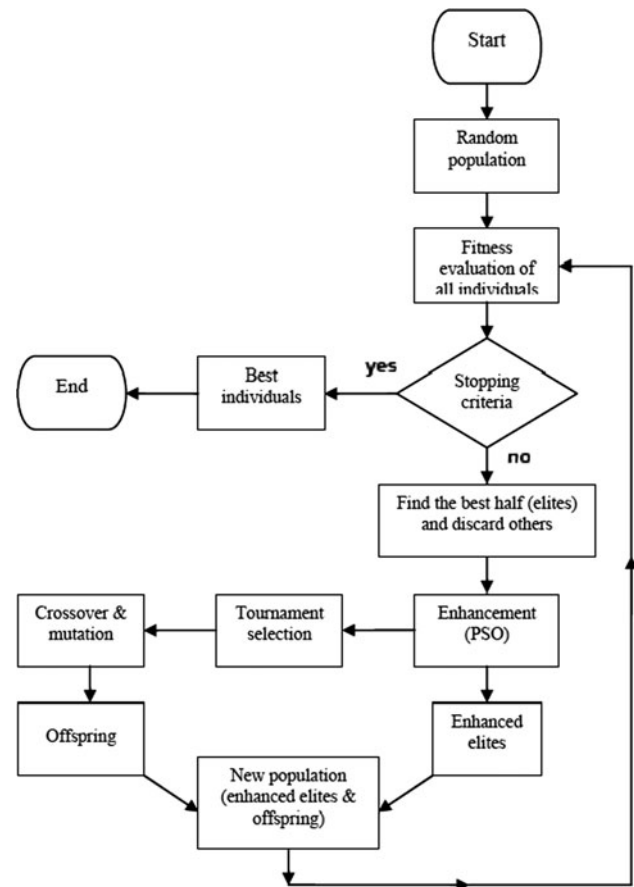


Fig. 3 Flowchart of HGAPSO

The remaining terms are problem-dependent parameters; for example, C_1 and C_2 represent “trust” parameters indicating how much confidence the current particle has in itself (C_1 or cognitive parameter) and how much confidence it has in the swarm (C_2 or social parameter), and ω is the inertia weight. The latter term plays an important role in the PSO convergence behavior since it is employed to control the exploration abilities of the swarm. It directly affects the current velocity, which in turn is based on the previous history of velocities. Large inertia weights allow for wide velocity updates providing the global exploration of the search space, while small inertia values concentrate the velocity updates to nearby regions of the design space.

4.2 Genetic algorithm

Genetic algorithm (GA) is a well-known and frequently used evolutionary computation technique. This method was originally developed by John Holland [20] and his PhD students Hassan et al. [21]. The idea was inspired from Darwin’s natural selection theorem that is based on the idea of the survival of the fittest. The GA is inspired by the

Table 1 T_{lopt} for different T_H and ε_R

T_H	ε_R											
	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95
650	645.81	645.8	645.75	645.74	645.72	645.7	645.65	645.61	645.58	645.53	645.46	645.31
700	675.12	675.02	674.9	674.76	674.62	674.45	674.26	674.04	673.79	673.49	673.15	672.72
750	705.1	704.91	704.69	704.45	704.18	703.87	703.53	703.13	702.67	702.12	701.49	700.7
800	735.81	735.53	735.22	734.87	734.48	734.03	733.53	732.96	732.29	731.5	730.57	729.44
850	767.33	766.96	766.55	766.09	765.58	765.28	764.34	763.59	762.71	761.69	760.47	758.98
900	799.69	799.24	798.73	798.17	797.54	796.83	796.02	795.09	794.01	792.75	791.24	789.41
950	832.96	832.42	831.82	831.15	830.4	829.56	828.6	827.5	826.23	824.72	822.94	820.77
1,000	867.15	866.53	865.84	865.07	864.21	863.24	862.13	860.87	859.39	857.66	855.6	853.09
1,050	902.3	901.6	900.82	899.95	898.98	897.88	896.63	895.2	893.53	891.57	889.24	886.4
1,100	938.4	937.62	936.75	935.79	934.71	933.49	932.1	930.51	928.66	926.48	923.88	920.73
1,150	975.44	974.59	973.64	972.58	971.4	970.06	968.54	966.79	964.76	962.37	959.52	956.05
1,200	1,013.4	1,012.5	1,011.5	1,010.3	1,009	1,007.6	1,005.9	1,004	1,001.8	999.23	996.13	992.36
1,250	1,052.3	1,051.3	1,050.2	1,049	1,047.6	1,046	1,044.2	1,042.2	1,039.8	1,037	1,033.7	1,029.6
1,300	1,092	1,090.9	1,089.8	1,088.5	1,087	1,085.3	1,083.4	1,081.3	1,078.7	1,075.7	1,072.2	1,067.8
1,350	1,132.6	1,131.4	1,130.2	1,128.8	1,127.2	1,125.5	1,123.5	1,121.2	1,118.5	1,115.3	1,111.6	1,106.9
1,400	1,173.9	1,172.7	1,171.4	1,169.9	1,168.3	1,166.4	1,164.3	1,161.9	1,159.1	1,155.7	1,151.8	1,146.9
1,450	1,215.9	1,214.7	1,213.3	1,211.8	1,210.1	1,208.1	1,205.9	1,203.4	1,200.4	1,196.9	1,192.7	1,187.6
1,500	1,258.6	1,257.3	1,255.9	1,254.3	1,252.5	1,250.5	1,248.2	1,245.5	1,242.5	1,238.8	1,234.5	1,229.1
1,550	1,302	1,300.6	1,299.1	1,297.5	1,295.6	1,293.5	1,291.1	1,288.4	1,285.2	1,281.4	1,276.9	1,271.3
1,600	1,345.9	1,344.5	1,343	1,341.3	1,339.3	1,337.2	1,334.7	1,331.8	1,328.5	1,324.6	1,319.9	1,314.1
1,650	1,390.3	1,388.9	1,387.3	1,385.6	1,383.6	1,381.3	1,378.8	1,375.8	1,372.4	1,368.4	1,363.5	1,357.6
1,700	1,435.3	1,433.8	1,432.2	1,430.4	1,428.3	1,426	1,423.4	1,420.4	1,416.8	1,412.7	1,407.7	1,401.6
1,750	1,480.7	1,479.2	1,477.5	1,475.6	1,473.5	1,471.2	1,468.5	1,465.3	1,461.7	1,457.4	1,452.3	1,446
1,800	1,526.5	1,524.9	1,523.2	1,521.3	1,519.2	1,516.7	1,514	1,510.8	1,507.1	1,502.7	1,497.4	1,491
1,850	1,572.6	1,571.1	1,569.3	1,567.4	1,565.2	1,562.7	1,559.9	1,556.6	1,552.8	1,548.3	1,543	1,536.4
1,900	1,619.2	1,617.6	1,615.8	1,613.8	1,611.6	1,609	1,606.2	1,602.8	1,599	1,594.4	1,588.9	1,582.2
1,950	1,666	1,664.4	1,662.6	1,660.6	1,658.3	1,655.7	1,652.8	1,649.4	1,645.4	1,640.7	1,635.1	1,628.3
2,000	1,713.2	1,711.5	1,709.7	1,707.6	1,705.3	1,702.7	1,699.7	1,696.2	1,692.2	1,687.4	1,681.7	1,674.7
2,050	1,760.6	1,758.9	1,757	1,754.9	1,752.6	1,749.9	1,746.9	1,743.4	1,739.3	1,734.4	1,728.6	1,721.5
2,100	1,808.2	1,806.5	1,804.6	1,802.5	1,800.1	1,797.4	1,794.3	1,790.8	1,786.6	1,781.7	1,775.8	1,768.6
2,150	1,856.1	1,854.4	1,852.4	1,850.3	1,847.9	1,845.1	1,842	1,838.4	1,834.2	1,829.2	1,823.2	1,815.9

principles of genetics and evolution and mimics the reproduction behavior observed in biological populations.

In GA, a candidate solution for a specific problem is called an individual or a chromosome and consists of a linear list of genes. GA begins its search from a randomly generated population of designs that evolve over successive generations (iterations), eliminating the need for a user supplied starting point. To perform its optimization like process, the GA employs three operators to propagate its population from one generation to another. The first operator is the “selection” operator in which the GA considers the principal of “survival of the fittest” to select and generate individuals (design solutions) that are adapted to their environment. The second operator is the “crossover” operator, which mimics mating in biological populations.

Table 2 Statistical properties of data set which used in this study

Parameters	Max	Min	Average
T_H	2,150	650	1,400
T_{lopt}	1,815.9	645.39	1,212.6489
ε_R	0.95	0.4	0.675

The crossover operator propagates features of good surviving designs from the current population into the future population, which will have a better fitness value on average. The last operator is “mutation,” which promotes diversity in population characteristics. The mutation operator allows for global search of the design space and prevents the algorithm from getting trapped in local minima [21].

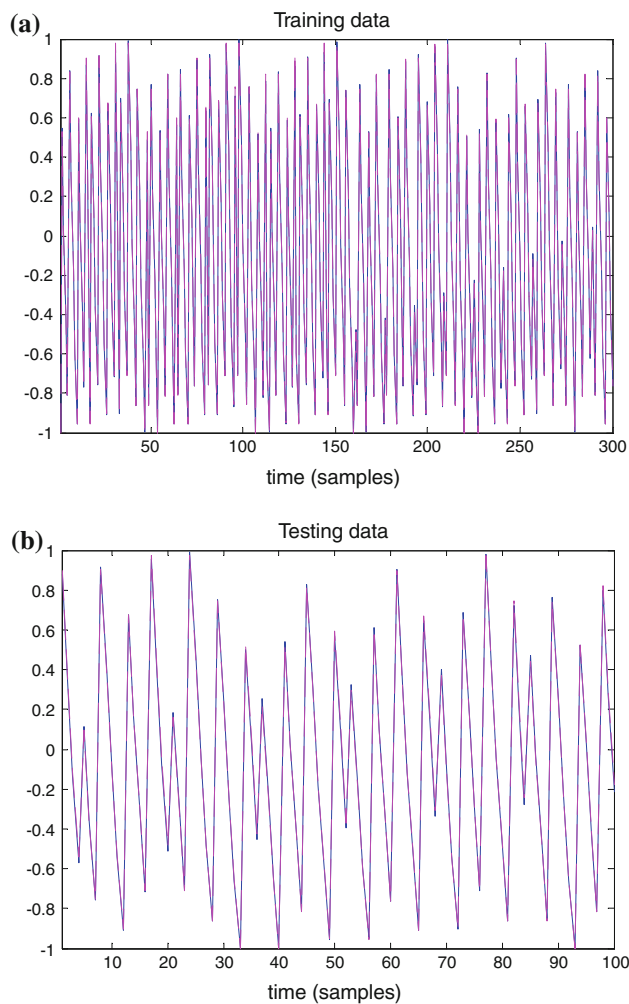


Fig. 4 Comparison between measured and predicted power (HGA-PSO-ANN): **a** Training **b** Testing

4.3 Hybrid genetic algorithm and particle swarm optimization (HGAPSO)

Although GAs have been successfully applied to a wide spectrum of problems, using GAs for large-scale optimization could be very expensive due to its requirement of a large number of function evaluations for convergence. This would result in a prohibitive cost for computation of function evaluations even with the best computational facilities available today. Considering the efficiency of the PSO and the compensatory property of GA and PSO, combining the searching abilities of both methods in one algorithm seems to be a logical approach. In this paper, the hybrid of GA and PSO named HGAPSO, originally presented by Juang [22], is used. The flowchart of the HGA-PSO is shown in Fig. 3.

It is obvious that the feasible region in constrained optimization problems may be of any shape (convex or concave and connected or disjointed). In real-parameter

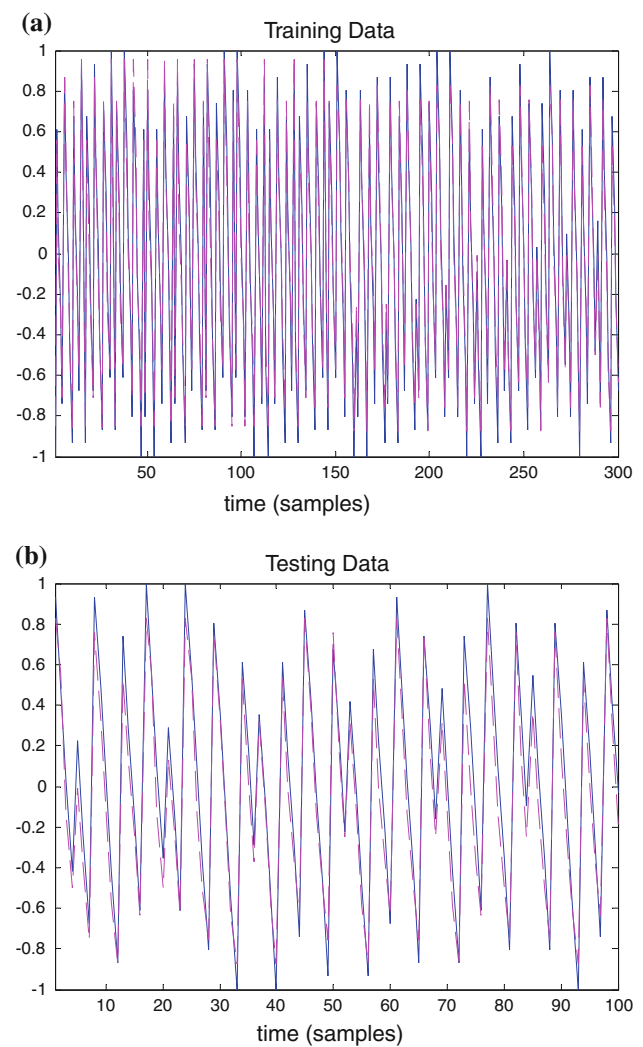


Fig. 5 Comparison between measured and predicted efficiency (HGAPSO-ANN): **a** Training **b** Testing

constrained optimization using GAs, schemata specifying contiguous regions in the search space (such as $110^*...^*$) may be considered to be more important than schemata specifying discrete regions in the search space (such as $(^*1^*10^*...^*)$, in general. Since any arbitrary contiguous region in the search space cannot be represented by single Holland's schema and since the feasible search space can usually be of any arbitrary shape, it is expected that the single-point crossover operator used in binary GAs will not always be able to create feasible children solutions from two feasible parent solutions.

The floating-point representation of variables in a GA and a search operator that respects contiguous regions in the search space may be able to eliminate the above two difficulties associated with binary coding and single-point crossover.

Hence, a floating-point coding scheme is adopted here for all of the GA, PSO, and HGAPSO. For the frame

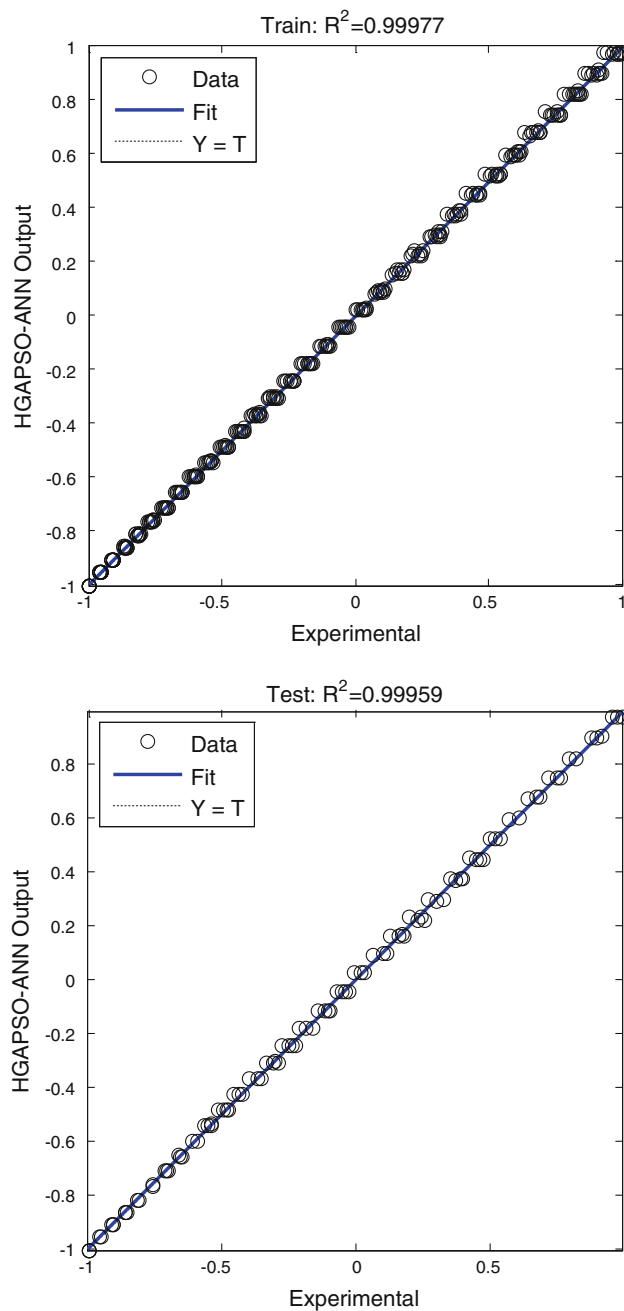


Fig. 6 R^2 HGAPSO-ANN power predicted

structures where design variables must have discrete values, the solutions are achieved by rounding the design variables to the nearest permissible integer number.

5 HGAPSO-ANN model results

In this study, an artificial neural network was used to build a model to predict the power of the solar stirling heat engine by using the data set reported in literature of china

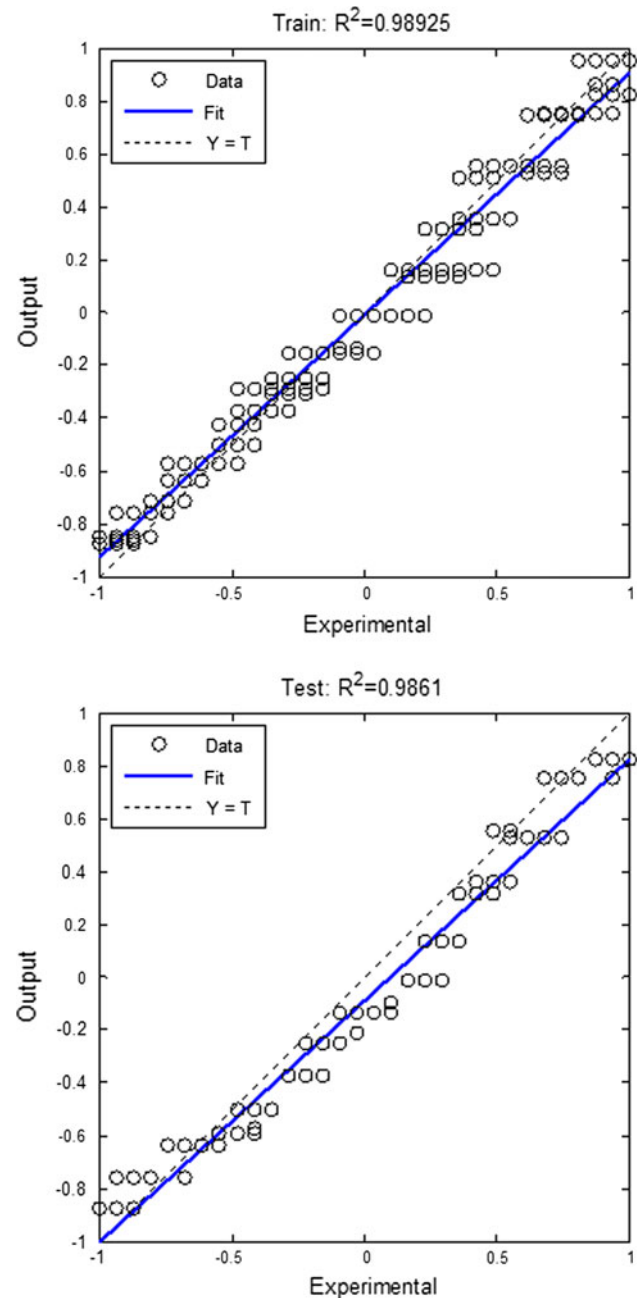


Fig. 7 R^2 HGAPSO-ANN efficiency predicted

[5]. Data set that is used in this study is reported in Table 1. Also, statistical properties of data set are shown in Table 2. The best ANN architecture was 3-4-10-1 (3 input units, 4 neuron in first hidden layer, 10 neuron in second hidden layer, and 1 output neuron). ANN model trained with back propagation network was trained by Levenberg–Marquardt to predict power of the solar stirling heat engine using three parameters (effectiveness of the regenerator, absorber temperature, and high temperature of working fluid) as inputs. The transfer functions in hidden and output layer are sigmoid and linear, respectively.

Table 3 Performances of HGAPSO-ANN model

	Power	Efficiency
MSE	0.0001994	0.010111
R^2	0.99959	0.9861

HGAPSO is used as neural network optimization algorithm and the mean square error (MSE) as a cost function in this algorithm. The goal in proposed algorithm is minimizing this cost function. Every weight in the network is initially set in the range of $[-1, 1]$, and every initial particle is a set of weights generated randomly in the range of $[-1, 1]$.

We used 300 data samples that were chosen by a random number generator for network training. The remaining 100 samples were put aside to be used for testing the network's integrity and robustness.

The power and efficiency of the solar stirling heat engine prediction in the training and test phase are shown in Figs. 4 and 5, respectively. The simulation performance of the HGAPSO-ANN model was evaluated on the basis of mean square error (MSE) and efficiency coefficient R^2 . Table 3 gives the MSE and R^2 values for HGAPSO-ANN model of the validation phases. In general, a R^2 value greater than 0.9 indicates a very satisfactory model performance, while a R^2 value in the range 0.8–0.9 signifies a good performance, and value less than 0.8 indicates an unsatisfactory model performance. Figures 6 and 7 show the extent of the match between the measured and predicted power and efficiency of the solar stirling heat engine values by HGAPSO-ANN model in terms of a scatter diagram.

6 Conclusions

In this article, we have presented a genetic algorithm evolved neural network. Our methodology presents a hybrid genetic algorithm and particle swarm optimization-based neural network (HGAPSO-ANN), which effectively combines the local searching ability of the hybrid particle swarm optimization and genetic algorithm. The idea of our algorithm is that each initial point of the neural network is selected by a hybrid genetic algorithm and particle swarm optimization, and the fitness of the hybrid genetic algorithm and particle swarm optimization is determined by neural network. The hybrid genetic algorithm and particle swarm parameters are carefully designed to optimize the neural network, avoiding premature convergence. The experimental measurement data have showed that the predictive performance of the proposed model is very well. This has been supported by the analysis of the changes of connection weights and biases of the neural network. One

problem when considering the combination of neural networks and hybrid genetic algorithm and particle swarm optimization for prediction of power and efficiency of the solar stirling heat engine is the determination of the optimal neural network topology. Our neural network topology described in this experiment is determined manually. A substitute method is to apply the particle swarm optimization for neural network structure optimization, which will be a part of our future work.

References

1. Thomas M, Peter H (2003) Dish-stirling systems: an overview development and status. *J solar Energy Eng* 125:135–151
2. Thombare DG, Verma SK (2008) Technological development in the stirling cycle engines. *Renew Sustain Energy Rev* 12:1–38
3. Luo YJ, He XN, Wang CG (2005) Solar application technology. Chemical Industry Press, Beijing, pp 253–271
4. Rallo R, Ferre-Gin J, Arenas A, Giral F (2002) Neural virtual sensor for the inferential prediction of product quality from process variables. *Comput Chem Eng* 26:1735–1754
5. Yaqi Li, Yaling He, Weiwei Wang (2011) Optimization of Solar-powered stirling heat engine with finite-time thermodynamics. *Renewable Energy* 36:421–427
6. Kim J-S, Kang YH (2006) Progress of R& D efforts for solar thermal power generation in Sout Korea. *SolarPACES 13th international symposium*, June 20–23 seville Spain, IEA SolarPACES
7. Wu F, Chen LG, Sun FR et al (2008) Performance optimization of stirling engine and cooler based on finite-time thermodynamic. Chemical Industry Press, Beijing, p 59
8. Costea M, Petrescu S, Harman C (1999) The effect of irreversibility's on solar stirling engine cycle performance. *Energy Convers Manage* 40:1723–1731
9. Kaushik SC, Kumar S (2000) Finite time thermodynamic analysis of end reversible stirling heat engine with regenerative losses. *Energy* 25:989–1003
10. Kaushik SC, Kumar S (2001) Finite time thermodynamic evaluation of irreversible Ericsson and stirling heat engines. *Energy Convers Manage* 42:295–312
11. Ahmet D, Oguz SS, Bahri S, Hasbi Y (2004) Optimization of thermal systems based on finite-time thermodynamics and thermo economics. *Prog Energy Combust Sci* 30:175–271
12. Hornik K, Stinchcombe M, White H (1990) Universal approximation of an unknown mapping and its derivatives using multi-layer feed forward networks. *Neural Net w* 3(5):551–560
13. Garcia-Pedrajas N, Hervas-Martinez C, Munoz-Perez J, COV-NET (2003) A cooperative co-evolutionary model for evolving artificial neural networks. *IEEE Trans Neural Netw* 14:575–596
14. Brown M, Harris C (1994) Neural fuzzy adaptive modeling and control. Prentice-Hall, Englewood Cliffs
15. Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: *Proceedings of the 6th international symposium on micro machine and human science*. Nagoya, Japan, pp 39–432
16. Eberhart RC, Simpson PK, Dobbins RW (1966) Computational intelligent PC tools. Academic Press Professional, Boston
17. Shi YH, Eberhart RC (1998) A modified particle swarm optimizer. In: *Proceedings of IEEE world conference on computation intelligence*, pp 69–73
18. Rechenberg I (1994) Evolution strategy, in computational intelligence: imitating life. IEEE Press, Piscataway

19. Schwefel HP (1995) Evolution and optimum seeking. Wiley, New York
20. Holland JH (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor
21. Hassan R, Cohanin B, Weck O (2005) A comparison of particle swarm optimization and the genetic algorithm. In: Proceedings of 46th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics & materials conference, Austin, Texas, pp 18–21
22. Juang CF (2004) A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. IEEE Trans Syst Man Cybern B Cybern 34(2):997–1006